

OMG Model Driven Architecture

**Arquitectura dirigida por
modelos**

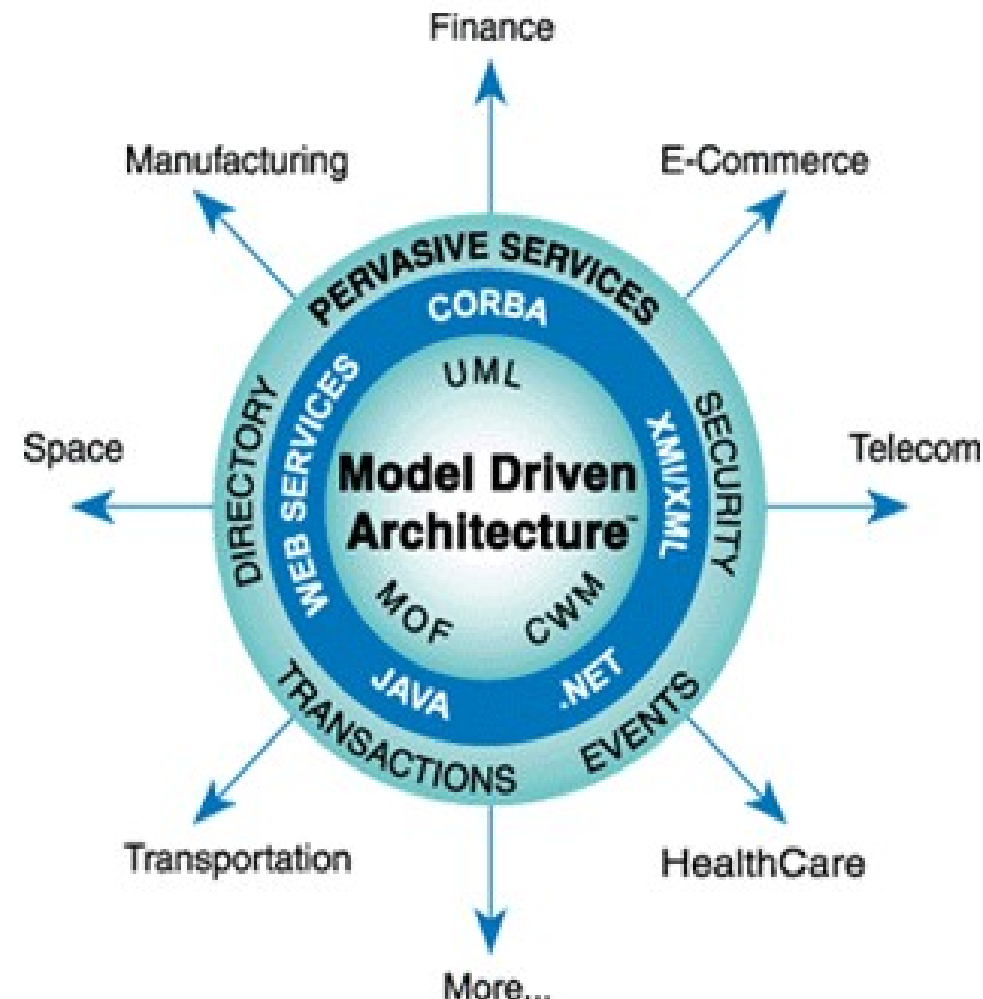
MDA: model driven architecture

Separa el negocio y la lógica de aplicación de la plataforma tecnológica subyacente

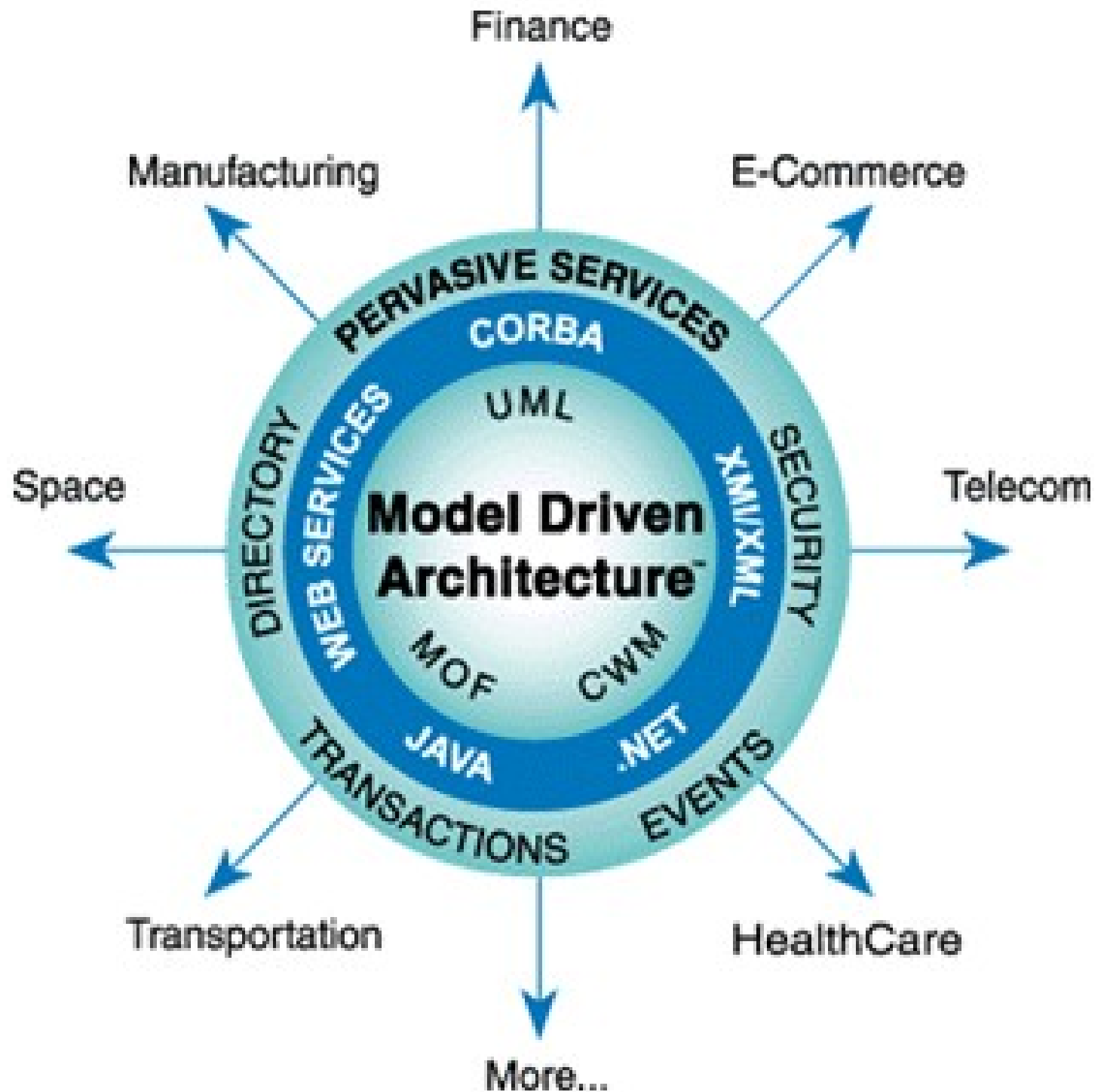


How Systems Will Be Built

OMG's Model Driven Architecture® (MDA®) provides an open, vendor-neutral approach to the challenge of business and technology change. Based on OMG's established standards, the MDA separates business and application logic from underlying platform technology. Platform-independent models of an application or integrated system's business functionality and behavior, built using UML and the other associated OMG modeling standards, can be realized through the MDA on virtually any platform, open or proprietary, including Web Services, .NET, CORBA®, J2EE, and others. These platform-independent models document the business functionality and behavior of an application separate from the technology-specific code that implements it, insulating the core of the application from technology and its relentless churn cycle while enabling interoperability both within and across platform boundaries. No longer tied to each other, the business and technical aspects of an application or integrated system can each evolve at its own pace - business logic responding to business need, and technology taking advantage of new developments - as the business requires.



MDA



MDA

- **UML** Unified Modeling Language
- **CWM** Common Warehouse Meta-model
 - estandariza esquemas de Bases de Datos,
transformaciones de esquemas ...
- **MOF** Meta-Object Facility
 - estandariza una facilidad para gestionar repositorios
de modelos

MDA

- CORBA

Common Object Request Broker
Architecture

estandar de interoperabilidad

- XMI XML Metadata Interchange

formato de intercambio para modelos basado en
MOF y XML

MDA

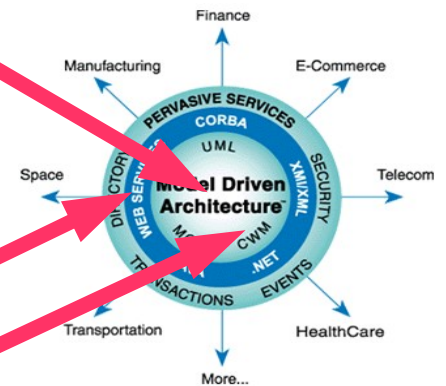
- El MDA es un enfoque para el desarrollo de sistemas ...que... ofrece una forma de que los modelos dirijan el proceso de comprensión, diseño, construcción, instalación, operación, mantenimiento y modificación

MDA

- El MDA es neutral respecto a:
 - Lenguaje de programación
 - Vendedor
 - middleware

MDA

- El MDA puede poseer varios (pocos) núcleos (core models)
 - Enterprise computing
 - Real-time
 - ...
- El núcleo está basado en los estándares de modelización UML, MOF, CWM
- El núcleo es independiente de la plataforma de middleware: CCM, EJB, COM+





Middleware

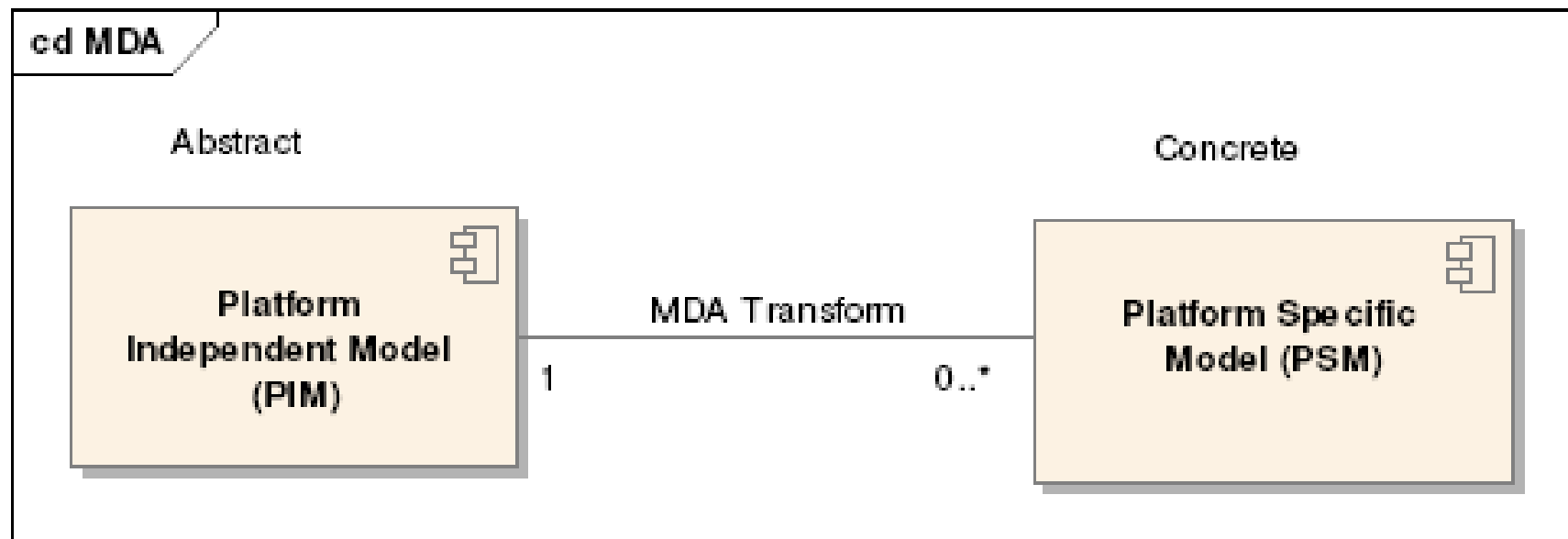
El **Middleware** es un **software** de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (**sistema operativo** y **red**). El Middleware nos abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipo de servicios de middleware.

Por lo general el middleware del lado cliente está implementado por el Sistema Operativo subyacente, el cual posee las librerías que implementan todas las funcionalidades para la comunicación a través de la red.

Aplicaciones basadas en MDA

1. Crear un modelo de aplicación general, independiente de la plataforma, en términos del núcleo, usando UML
2. Un experto en la plataforma elegida (CCM, EJB, COM+) convierte el modelo general en uno adaptado a la misma:
 - Modelo UML adaptado, expresado en un dialecto (perfil) para la plataforma
 - Elementos específicos de la plataforma (IDL, XML, ...)
3. Generar el código de la aplicación

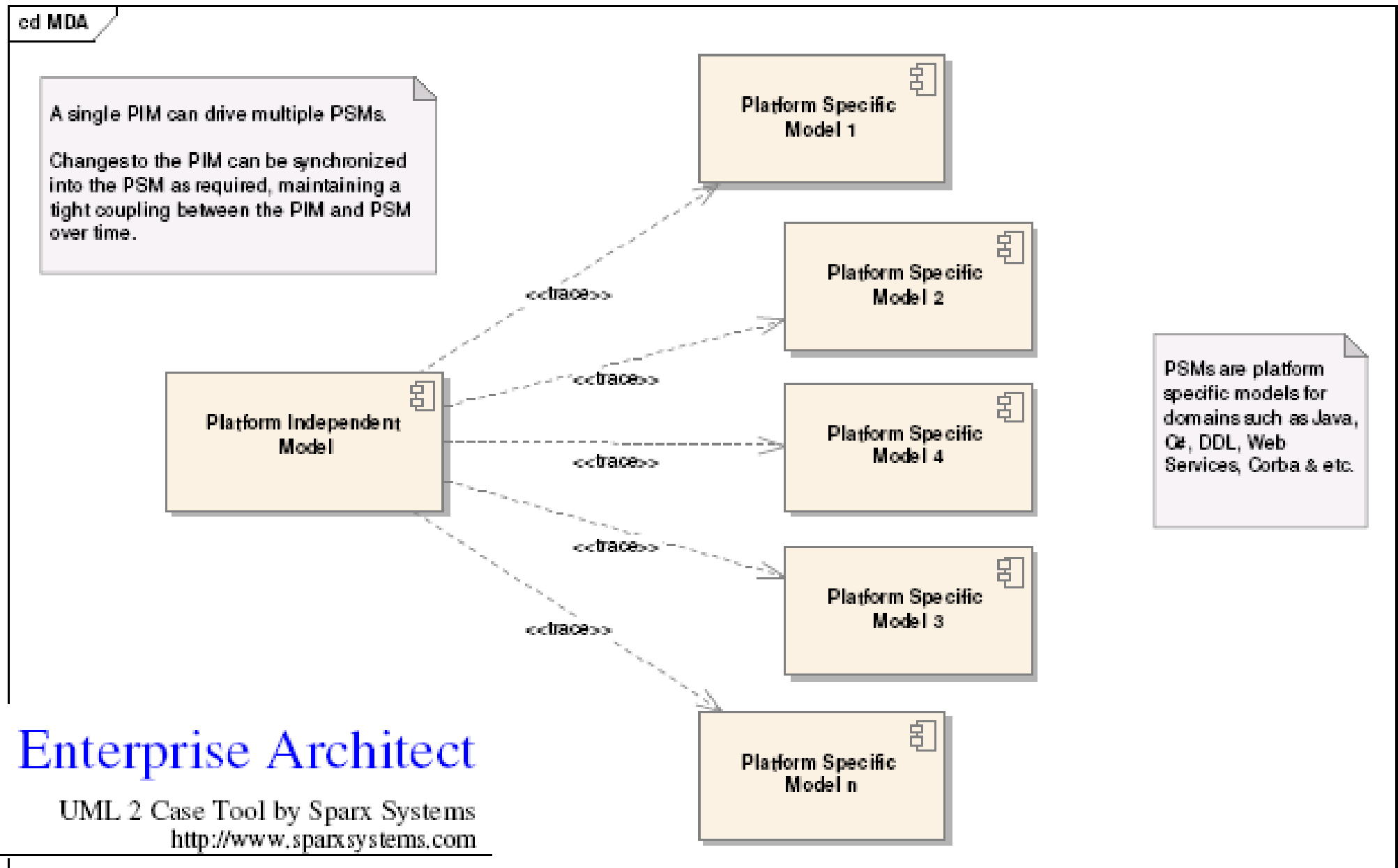
Aplicaciones basadas en MDA



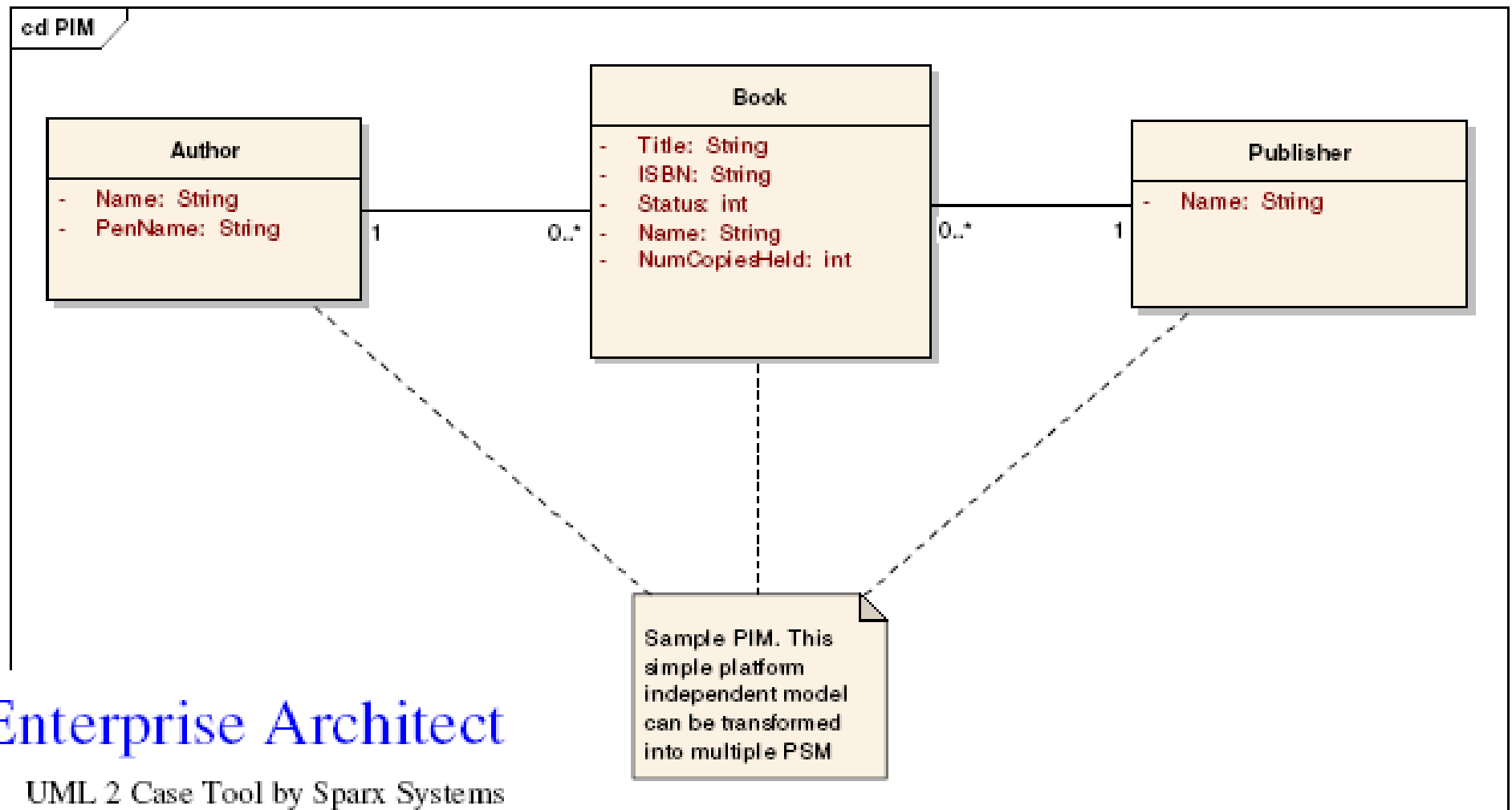
Enterprise Architect

UML 2 Case Tool by Sparx Systems
<http://www.sparxsystems.com>

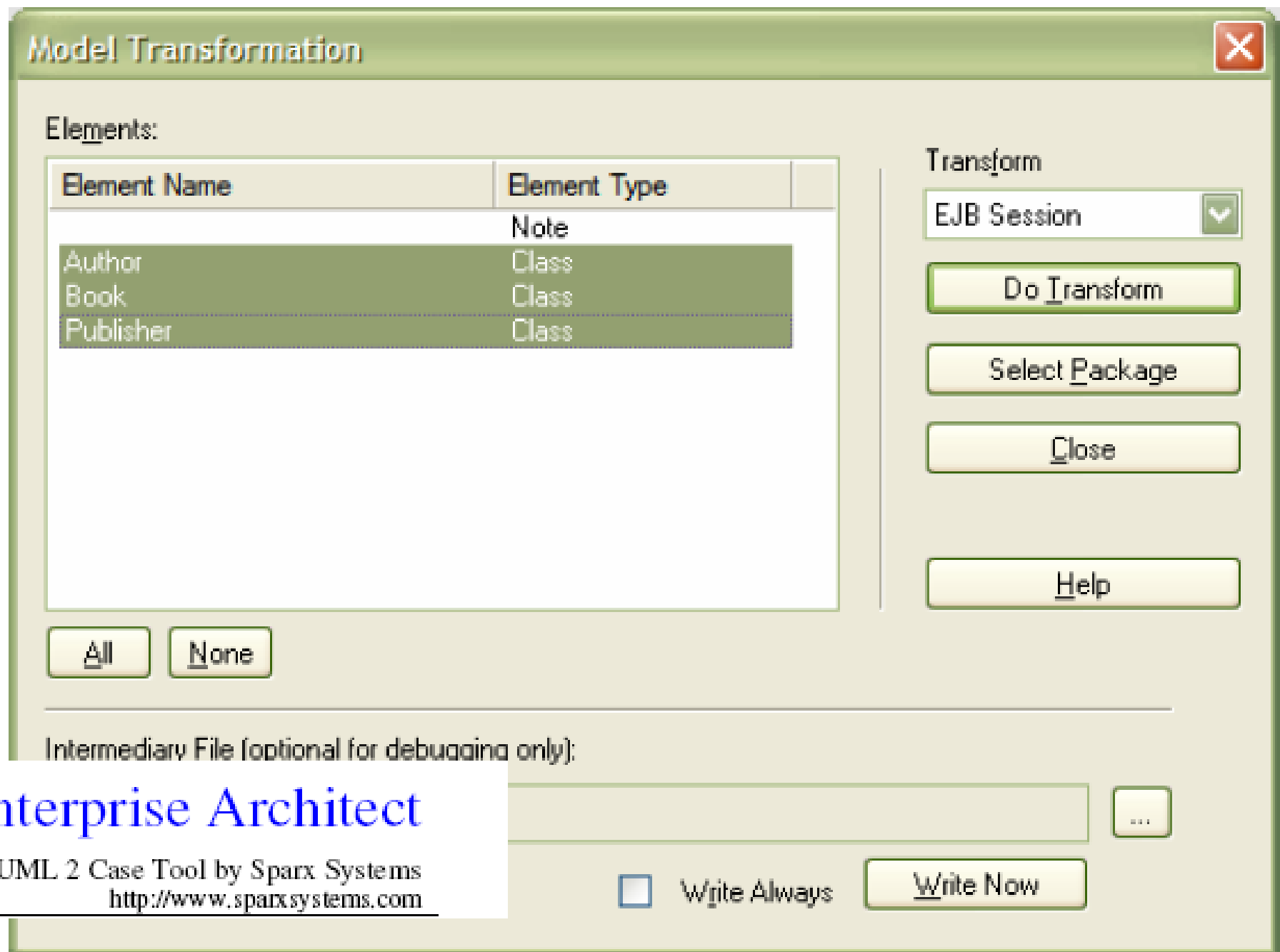
Aplicaciones basadas en MDA



Aplicaciones basadas en MDA



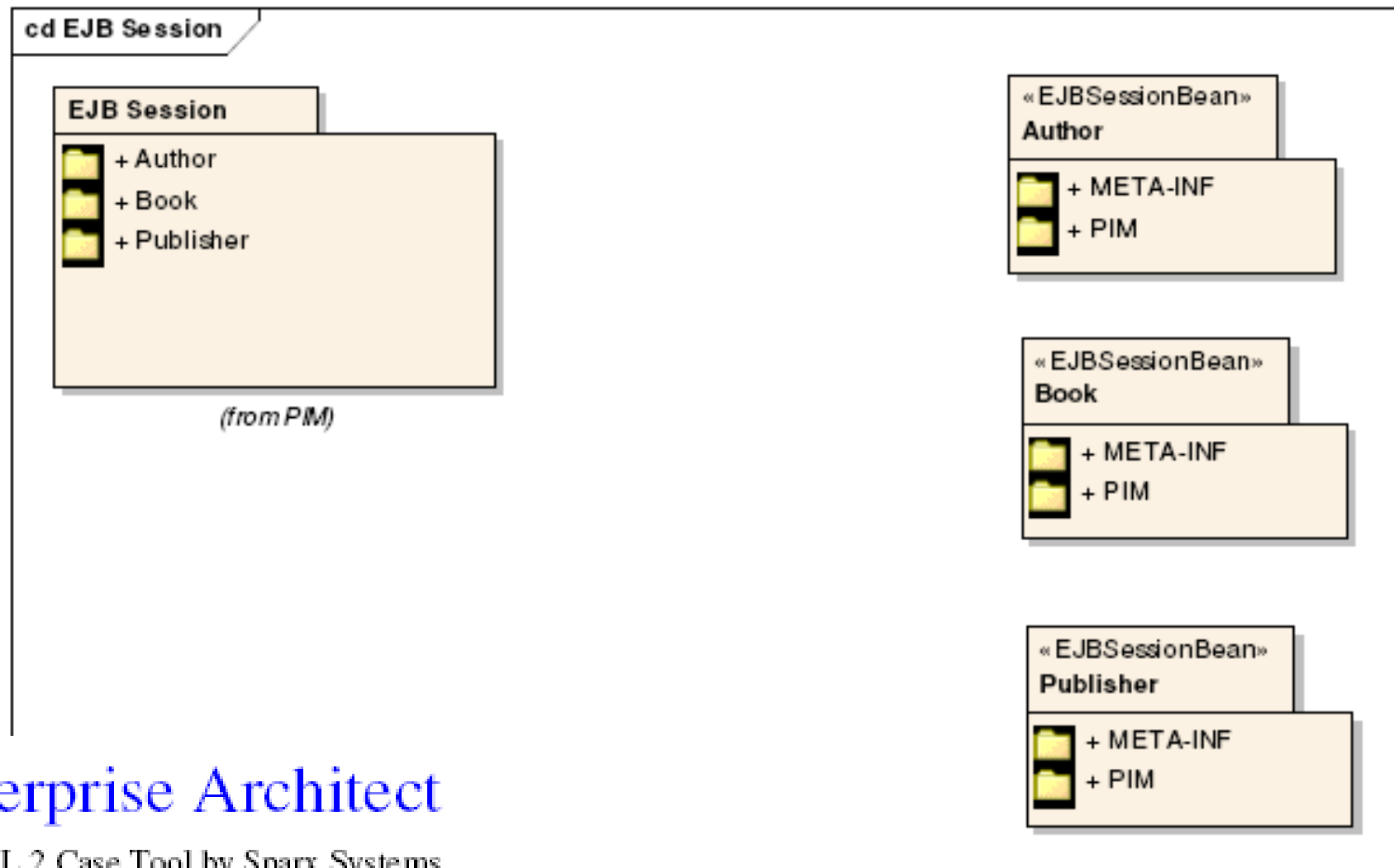
Aplicaciones basadas en MDA



Enterprise Architect

UML 2 Case Tool by Sparx Systems
<http://www.sparxsystems.com>

Aplicaciones basadas en MDA



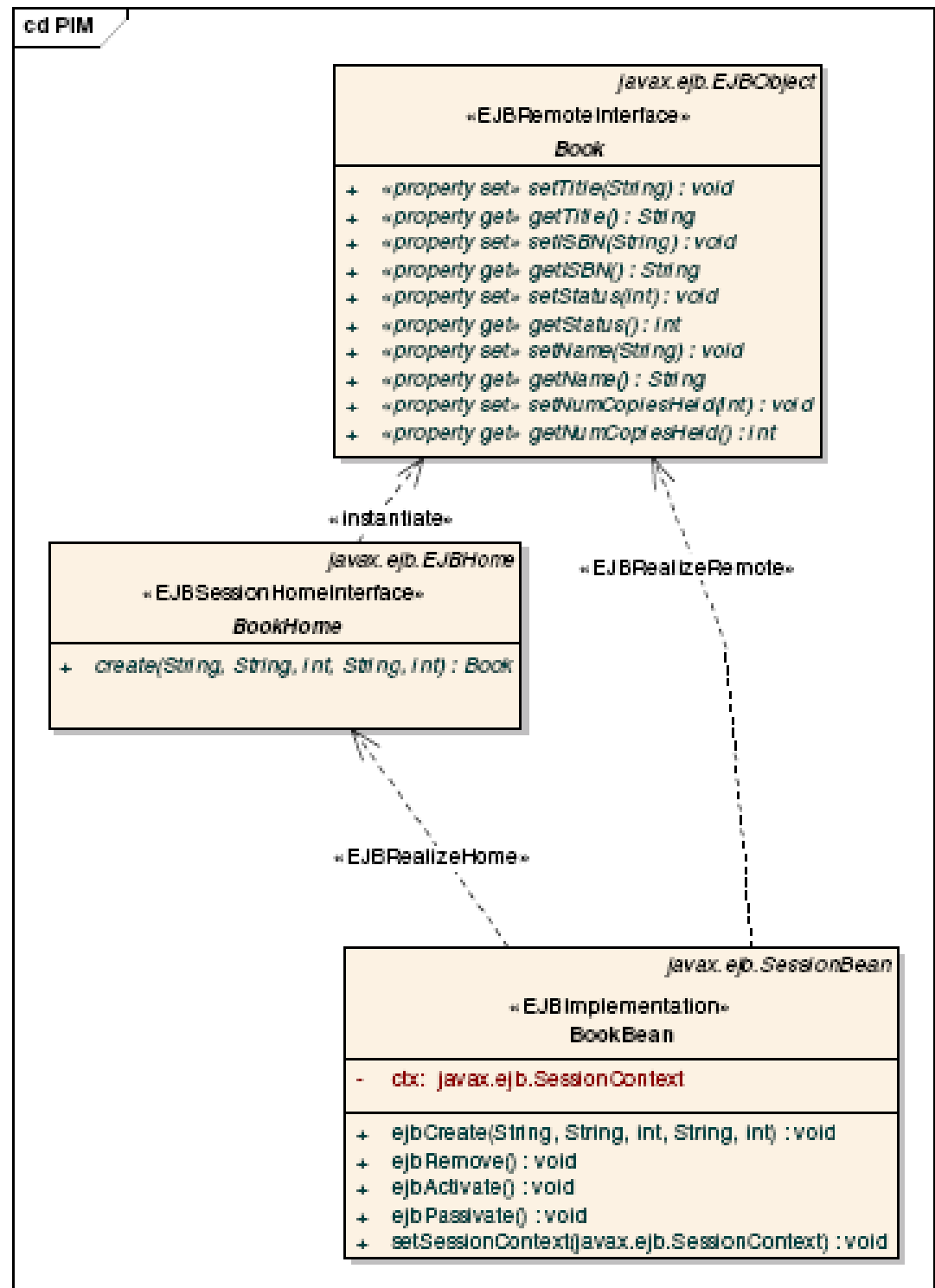
Enterprise Architect

UML 2 Case Tool by Sparx Systems
<http://www.sparxsystems.com>

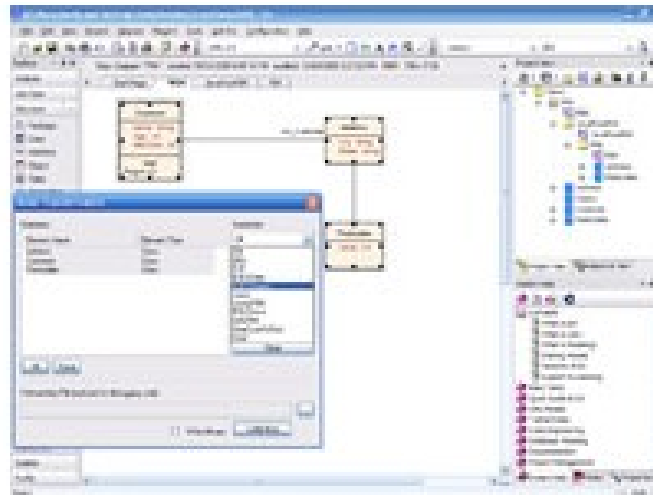
Aplicaciones basadas en MDA

Enterprise Architect

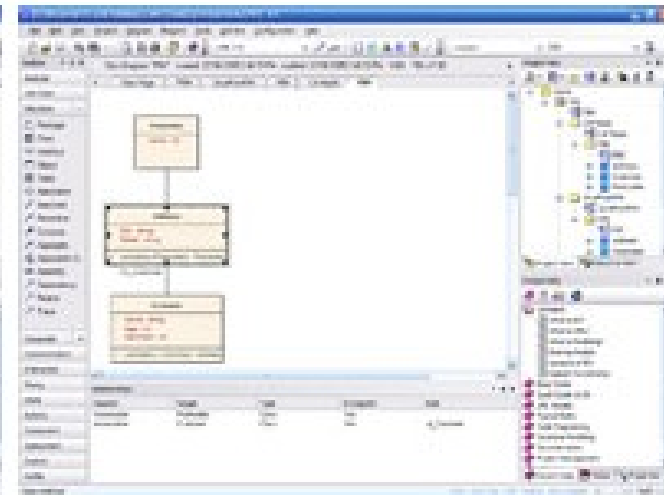
UML 2 Case Tool by Sparx Systems
<http://www.sparxsystems.com>



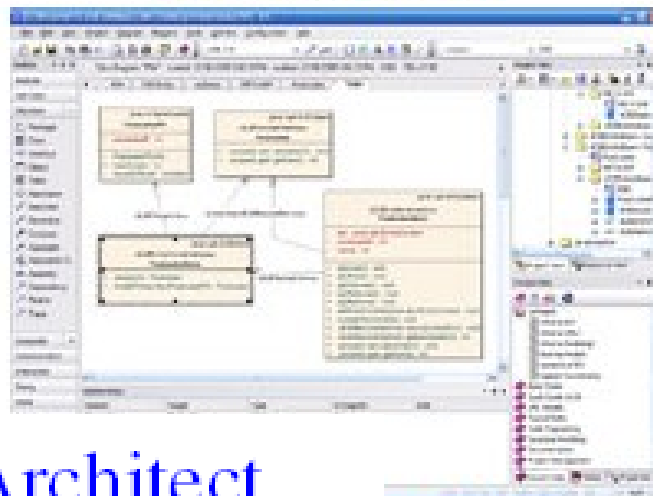
Aplicaciones basadas en MDA



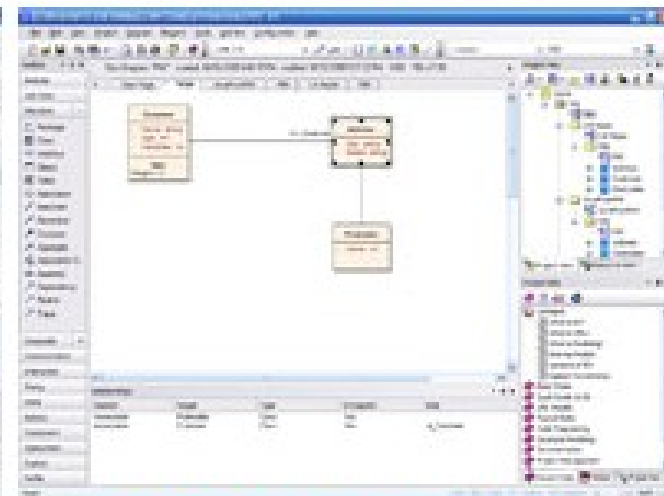
MDA Transform



MDA Transform - C#



MDA Transform - EJB



MDA Transform - PIM

Enterprise Architect

UML 2 Case Tool by Sparx Systems
<http://www.sparxsystems.com>

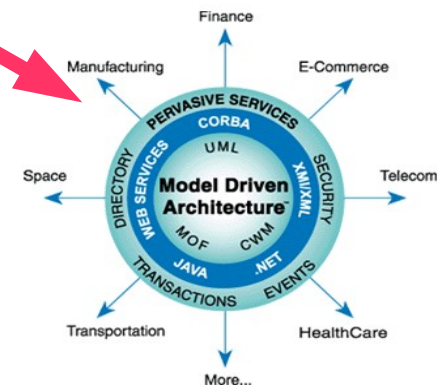
Ventajas de esta arquitectura

- La semántica esencial de la aplicación se ha elaborado de forma independiente de la plataforma
- Se posibilita la definición de modelos estandar de dominios
- Podrán definirse correspondencias entre el núcleo y las nuevas plataformas de middleware, que faciliten la migración de aplicaciones
- Los modelos
 - Se construyen y expresan en UML
 - Se transmiten en XMI
 - Se almacenan en repositorios MOF
- La formalización de los modelos incrementará la calidad de los diseños



DTF: Domain Task Forces

- El esfuerzo de estandarización del OMG se apoya en diferentes DTFs, correspondiente a diversos campos de aplicación:
 - Finanzas, Transportes, Salud, Telecomunicaciones, Espacial, Fabricación, E-comercio, ...
- La estandarización se producirá a distintos niveles
 - Modelos independientes de la plataforma
 - Modelos dependientes de la plataforma
 - Interfaces de componentes
 - XML DTD



Servicios MDA

- Las aplicaciones hacen uso de servicios esenciales:
 - Servicios de directorio, Persistencia, Procesamiento transaccional, Gestión de eventos, ...
- O poseen ciertos atributos hardware o software:
 - Tolerancia a fallos, Escalabilidad, Sistemas empotrados, ...
- OMG pretende especificar estos servicios, primero, de forma independiente de la plataforma, para generar posteriormente definiciones específicas

